



TITLE:

# OpenXM 1.1.3の概要 (数式処理における理論と応用の研究)

AUTHOR(S):

小原, 功任; 高山, 信毅; 田村, 恭士; 野呂, 正行; 前川, 将秀

---

CITATION:

小原, 功任 ...[et al]. OpenXM 1.1.3の概要 (数式処理における理論と応用の研究). 数理解析研究所講究録 2001, 1199: 179-191

ISSUE DATE:

2001-04

URL:

<http://hdl.handle.net/2433/64915>

RIGHT:

## OpenXM 1.1.3 の概要

金沢大学理学部	小原 功任(Katsuyoshi OHARA) *
神戸大学理学部	高山 信毅(Nobuki TAKAYAMA) †
神戸大学大学院自然科学研究科	田村 恭士(Takashi TAMURA) ‡
神戸大学理学部	野呂 正行(Masayuki NORO) §
神戸大学大学院自然科学研究科	前川 将秀(Masahide MAEKAWA) ¶

### 1 OpenXM とは

OpenXM は, おなじタイプまたは異なるタイプの数学プロセス間でメッセージを交換するための規約である. 数学プロセス間でメッセージをやりとりすることにより, ある数学プロセスから他の数学プロセスを呼び出して計算を行なったり, 他のマシンで計算を行なわせたりすることが目的である. 開発の動機は, 手作り (または研究的な) 数学ソフトウェアの相互乗り入れの実現および分散計算の実装が第一であったが, もちろん数学ソフトウェア間だけでなく, ワープロソフトウェアや, インタラクティブな数学本, さらに数学デジタル博物館用のソフトウェアがこの規約に従い, 数学ソフトウェアを呼び出すことなどにも利用できる. 当面の目標は OpenXM 数学ソフトパッケージを作ることである. これはさまざまな数学ソフトウェアを一つのパッケージとして簡単に好きな言語より使えるようにするプロジェクトである. 現在 OpenXM.tar.gz には, asir, sm1, phc, gnuplot, tigers が入っている. OpenXM プロジェクトでは数ヵ月おきに安定版のソフトウェアをリリースしていて, 2001 年 1 月現在のバージョンは 1.1.3 である.

この論文では, 特に TCP/IP を用いた実装に準拠して OpenXM の説明を行う.

---

\*ohara@kappa.s.kanazawa-u.ac.jp

†takayama@math.sci.kobe-u.ac.jp

‡tamura@math.sci.kobe-u.ac.jp

§noro@math.sci.kobe-u.ac.jp

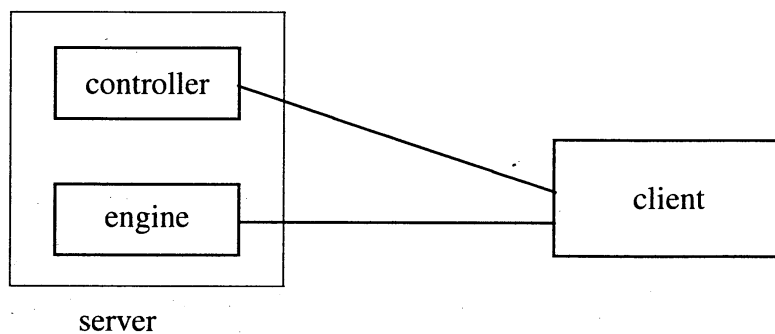
¶maekawa@math.sci.kobe-u.ac.jp

## 2 OpenXM の計算のモデル

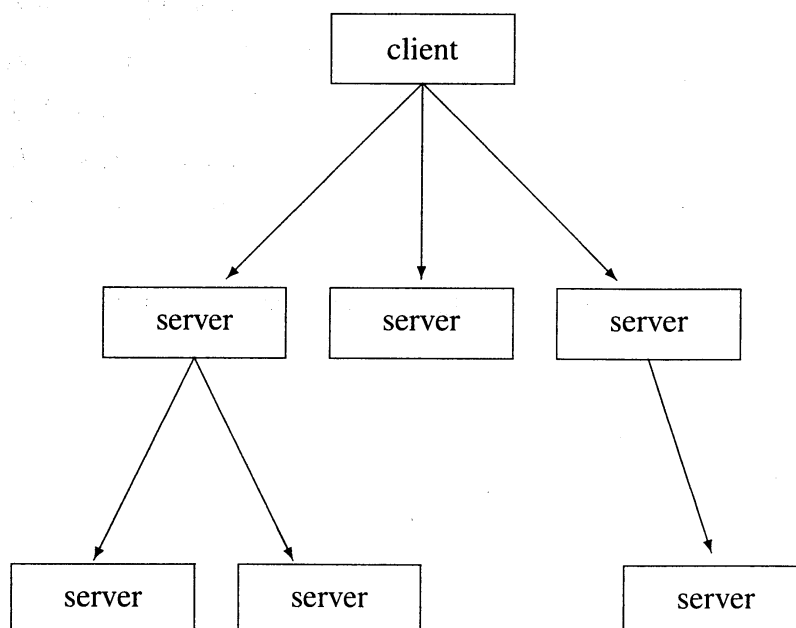
ここでは OX-RFC-100 [6] で規定された計算の仕組みについて説明する。

まず, OpenXM のソフトウェアはクライアント・サーバモデルをとっている. さらにサーバはコントローラとエンジンの二つのプロセスに分かれている. クライアントとエンジンは同一のマシン上になくてもよいが, コントローラとエンジンは同一のマシン上になければならない.

これらのクライアントとサーバの間でメッセージを交換することで計算を進行させるのが, OpenXM の考え方である. クライアントはエンジンに対してメッセージを送り, 計算を行わせる. 計算が終わったら, 計算結果を送り返させるように指示するメッセージをエンジンに送ると, エンジンはメッセージをクライアントに送り, 結果を返す. メッセージの構造や通信路の構成など, 規約の詳細については, ここでは解説しないが, それらについては [6] をご覧になられたい.



さらに, サーバが新たにクライアントになることも許される. したがって, OX-RFC-100 [6] では, 各数学プロセスが次のような木構造をなす, 分散・並列計算モデルを定義したことになる.



### 3 OX サーバと提供される数学関数

OpenXM の各サーバおよびその数学的機能のうち幾つかを例をあげて説明しよう.

#### 3.1 ox\_asir

Risa/Asir は Free で配布されている汎用数式処理ソフトウェアである. たとえば次のような機能を持つ.

1.  $\mathbf{Q}$  係数の  $n$  変数多項式の因数分解 (関数 fctr).

例:

```
fctr(y^5-4*y^4+(-x^2+2*x+3)*y^3-x^2*y^2+4*x^2*y+x^4-2*x^3-3*x^2);
[[1,1],[y^3-x^2,1],[y-x-1,1],[y+x-3,1]]
```

2.  $\mathbf{Q}$  の代数拡大体における 1 変数多項式の因数分解. (関数 af)

次の例は多項式  $x^6 - 1$  を  $\mathbf{Q}$  に  $x^2 + x + 1 = 0$  の根を添加した体で因数分解する例である.

```
[374] load("gr")$ load("sp")$
[375] A1=newalg(x^2+x+1);
(#0)
[476] af(x^6-1,[A1]);
[[1,1],[x+(#0+1),1],[x+(#0),1],[x-1,1],[x+(-#0),1],
[x+1,1],[x+(-#0-1),1]]
```

$x^2 + x + 1 = 0$  の根を  $\omega$  としよう. ( $\omega^3 = 1, \omega + 1 = -\omega^2$  である.) 上の結果は,  $\omega$  を用いると,

$$x^6 - 1 = (x + \omega + 1)(x + \omega)(x - 1)(x - \omega)(x + 1)(x - \omega - 1)$$

と因数分解されることを示している.

3. 多項式環および微分作用素環 (ワイル代数) におけるグレブナ基底計算 (関数 gr および dp\_gr\_weyl\_main). 応用として準素イデアル分解の機能ももつ (関数 primadec). 例として変数の消去をグレブナ基底の計算でおこなってみよう.  $I$  を  $n$  変数多項式環  $\mathbf{Q}[x_1, \dots, x_n]$  のイデアルとすると  $I \cap \mathbf{Q}[x_1]$  の生成元は,  $I$  のグレブナ基底を辞書式順序 (lexicographic order)  $x_n > \dots > x_2 > x_1$  で計算して, グレブナ基底の中から,  $x_1$  のみの多項式を取り出してくれば求まる. たとえば,  $I$  として,  $x_1^2 + x_2^2 - 4, x_1 x_2 - 1$  で生成されるイデアルを考える.

```
[375] load("gr")$
[461] gr([x1^2+x2^2-4,x1*x2-1],[x2,x1],2);
[-x1^4+4*x1^2-1,x2+x1^3-4*x1]
```

$-x_1^4 + 4x_1^2 - 1$  が,  $I \cap \mathbf{Q}[x_1]$  の生成元である. 消去法とグレブナ基底との関連については, たとえば Cox, Little, O'Shea の教科書 [1] の 2 章, 3 章に入門的な説明がある.

Asir が用いている数学アルゴリズムについては, 野呂による解説 [5] を参照.

### 3.2 ox\_sml

Kan/sml および Kan/k0 は微分作用素環 (ワイル代数)  $D$  におけるグレブナ基底計算をもとにして,  $D$  加群の種々の構成や, 代数多様体のコホモロジの計算をおこなう. 現在, 因数分解, 準素イデアル分解,  $b$ -関数の  $D$  での計算などは, OpenXM プロトコル (OX-RFC 100, 101) を利用して, ox\_asir が担当しており, ox\_sml はサーバであるのみならず, asir の OpenXM サーバ機能をフルに利用しているクライアントでもある. Kan はたとえば次のような機能をもつ.

1.  $D$  でのグレブナ基底計算 (関数 gb).

グレブナ基底計算の応用として, あたえられた連立線形偏微分方程式系の解空間の次元の計算 (holonomic rank) がある. これを例として挙げておこう.

```
sml>(cohom.sml) run ;
sml> [ [( x*Dx + y*Dy -3 ) ( Dx-Dy )] (x,y)] rank ::
1
```

この出力は微分方程式系

$$\left[ x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y} - 3 \right] f = \left[ \frac{\partial}{\partial x} - \frac{\partial}{\partial y} \right] f = 0$$

の解空間の次元は 1 であることを意味する. この場合は実際に解を書き出すことができ,  $f = (x + y)^3$  が解である.

2. 代数多様体, たとえば  $\mathbf{C}^n \setminus V(f)$  のコホモロジ群の計算 (関数 deRham).  
たとえば Kan/k0 での計算は次のようになる.

```
% cd OpenXM/src/k097/lib/restriction
% k0
This is kan/k0 Version 1998,12/15
```

```

sm1 version = 3.001203
Default ring is Z[x,h].
In(2)= load("demo.k");
In(3)= DeRham2WithAsir( x^3-y^2 ):
Step1: Annihilating ideal (II)
      [      -3*x^2*Dy-2*y*Dx , -2*x*Dx-3*y*Dy-6 ]
Step2: (-1,1)-minimal resolution (Res0) [
      [
      [      2*x*Dx+3*y*Dy-h^2 ]
      [      -3*y*Dx^2+2*x*Dy*h ]
      ]
      [
      [      3*y*Dx^2-2*x*Dy*h , 2*x*Dx+3*y*Dy ]
      ]
      ]
Step3: computing the cohomology of the truncated complex.
Roots and b-function are
      [      [      1 ] , [      9*s^3-18*s^2+11*s-2 ] ]

[      [      0 , 1 , 1 ] ,
[      [      0 , [      ] ] , [      1 , [      ] ] ,
      [      1 , [      ] ] ] ]

```

Step1 の  $1/(x^3 - y^2)$  の満たす最大の微分方程式の計算, Step3 の一般化された  $b$ -関数とその根の計算では `ox_asir` を呼び出して計算している.

### 3. $D^m/I$ の $(u, v)$ -minimal free resolution (関数 `Sminimal`).

$D$  加群のアルゴリズムのアルゴリズムについては, 本 [4] が入門的話題から最先端の話題までを扱っている.

## 3.3 PHC

PHC pack は Jan Verschelde により開発された, 多面体ホモトピー法により代数方程式系の数値解を求めるシステムである. ホモトピー法というのは, たとえば代数方程式  $x^n - x^3 + x^2 - 5 = 0$  を解くために, まずスタートシステム  $x^n - 5 = 0$  を解き, その解を連続的にもとの方程式の解に延長する方法である. 連立代数方程式系の場合には, よい性質をもつス

タートシステムを作る必要があり, そのために多面体の分割をもちいる. これが多面体ホモトピー法である. 本 [2] の 7.5 節にこの方法の基礎となる多面体の幾何と根の数の勘定に関する説明がある. 多面体ホモトピー法については, Verschelde による解説の他, 論文 [3] などが読みやすい.

### 3.4 gnuplot, M2, tigers, pari, mathematica, OMproxy

長さの関係もあり詳しくは紹介できないので, 表題のソフトウェアを簡単に紹介しよう.

1. gnuplot はグラフを作成するソフトウェア.
2. M2 (Macaulay 2) は D.Grayson と M.Stillman により開発されている, 計算代数幾何用のソフトウェア.
3. tigers は B.Hubert により開発された, affine toric variety の全てのグレブナ基底を求めるソフトウェア.
4. pari は A. Cohen らにより開発されている, 整数論用のソフトウェア.
5. Mathematica は有名なので説明の必要はないであろう.
6. OMproxy は Java でかかれた, OpenXM のクラスライブラリおよび OpenMath とのデータの相互変換のためのソフトウェアである.

### 3.5 1077 functions are available on our servers and libraries

OX server の提供する関数にどのようなものがあるか, リストと簡単なデモを掲載しておこう. これらの関数のなかの一部については, 前節までで簡単に説明した. より詳しくは各コンポーネントシステムのマニュアルや参照されている論文, 本を見る必要がある.

#### Operations on Integers

idiv, irem (division with remainder), ishift (bit shifting), iand, ior, ixor (logical operations), igcd, (GCD by various methods such as Euclid's algorithm and the accelerated GCD algorithm), fac (factorial), inv (inverse modulo an integer), random (random number generator by the Mersenne twister algorithm).

#### Ground Fields

Arithmetics on various fields: the rationals  $\mathbb{Q}$ ,  $\mathbb{Q}(\alpha_1, \alpha_2, \dots, \alpha_n)$  ( $\alpha_i$  is algebraic over  $\mathbb{Q}$ ),  $GF(p)$  ( $p$  is a prime of arbitrary size),  $GF(2^n)$ .

### Operations on Polynomials

sdiv , srem (division with remainder), ptozp (removal of the integer content), diff (differentiation), gcd (GCD over the rationals), res (resultant), subst (substitution), umul (fast multiplication of dense univariate polynomials by a hybrid method with Karatsuba and FFT+Chinese remainder), urembymul\_precomp (fast dense univariate polynomial division with remainder by the fast multiplication and the precomputed inverse of a divisor),

### Polynomial Factorization

fctr (factorization over the rationals), fctr\_ff (univariate factorization over finite fields), af (univariate factorization over algebraic number fields), sp (splitting field computation).

### Groebner basis

dp\_gr\_main (Groebner basis computation of a polynomial ideal over the rationals by the trace lifting), dp\_gr\_mod\_main (Groebner basis over small finite fields), tolex (Modular change of ordering for a zero-dimensional ideal), tolex\_gsl (Modular rational univariate representation for a zero-dimensional ideal), dp\_f4\_main ( $F_4$  over the rationals), dp\_f4\_mod\_main ( $F_4$  over small finite fields).

### Ideal Decomposition

primedec (Prime decomposition of the radical), primadec (Primary decomposition of ideals by Shimoyama/Yokoyama algorithm).

### Quantifier Elimination

qe (real quantifier elimination in a linear and quadratic first-order formula), simpl (heuristic simplification of a first-order formula).

```
[0] MTP2 = ex([x11,x12,x13,x21,x22,x23,x31,x32,x33],
x11+x12+x13 @== a1 @&& x21+x22+x23 @== a2 @&& x31+x32+x33 @== a3
@&& x11+x21+x31 @== b1 @&& x12+x22+x32 @== b2 @&& x13+x23+x33 @== b3
@&& 0 @<= x11 @&& 0 @<= x12 @&& 0 @<= x13 @&& 0 @<= x21
@&& 0 @<= x22 @&& 0 @<= x23 @&& 0 @<= x31 @&& 0 @<= x32 @&& 0 @<= x33)$
[1] TSOL= a1+a2+a3@=b1+b2+b3 @&& a1@>=0 @&& a2@>=0 @&& a3@>=0
@&& b1@>=0 @&& b2@>=0 @&& b3@>=0$
[2] QE_MTP2 = qe(MTP2)$
[3] qe(all([a1,a2,a3,b1,b2,b3],QE_MTP2 @equiv TSOL));
@true
```

### Visualization of curves

plot (plotting of a univariate function), ifplot (plotting zeros of a bivariate polynomial), conplot (contour plotting of a bivariate polynomial function).



**Miscellaneous functions**

det (determinant), qsort (sorting of an array by the quick sort algorithm), eval (evaluation of a formula containing transcendental functions such as sin, cos, tan, exp, log) roots (finding all roots of a univariate polynomial), lll (computation of an LLL-reduced basis of a lattice).

 **$D$ -modules** ( $D$  is the Weyl algebra)

gb (Gröbner basis), syz (syzygy), annfs (Annihilating ideal of  $f^s$ ), bfunction, schreyer (free resolution by the Schreyer method), vMinRes (V-minimal free resolution), characteristic (Characteristic variety), restriction in the derived category of  $D$ -modules, integration in the derived category, tensor in the derived category, dual (Dual as a  $D$ -module), slope.

**Cohomology groups**

deRham (The de Rham cohomology groups of  $\mathbb{C}^n \setminus V(f)$ , ext (Ext modules for a holonomic  $D$ -module  $M$  and the ring of formal power series).

**Differential equations**

Helping to derive and prove combinatorial and special function identities, gkz (GKZ hypergeometric differential equations), appell (Appell's hypergeometric differential equations), indicial (indicial equations), rank (Holonomic rank), rrank (Holonomic rank of regular holonomic systems), dsolv (series solutions of holonomic systems).

**OpenMATH support**

om\_xml (CMO to OpenMATH XML), om\_xml\_to\_cmo (OpenMATH XML to CMO).

**Homotopy Method**

phc (Solving systems of algebraic equations by numerical and polyhedral homotopy methods).

**Toric ideal**

tigers (Enumerate all Gröbner basis of a toric ideal. Finding test sets for integer program), arithDeg (Arithmetic degree of a monomial ideal), stdPair (Standard pair decomposition of a monomial ideal).

**Communications**

ox\_launch (starting a server), ox\_launch\_nox, ox\_shutdown, ox\_launch\_generic, generate\_port, try\_bind\_listen, try\_connect, try\_accept, register\_server, ox\_rpc, ox\_cmo\_rpc, ox\_execute\_string, ox\_reset (reset the server), ox\_intr, register\_handler, ox\_push\_cmo, ox\_push\_local, ox\_pop\_cmo, ox\_pop\_local, ox\_push\_cmd, ox\_sync, ox\_get, ox\_pops, ox\_select, ox\_flush, ox\_get\_serverinfo

In addition to these functions, Mathematica functions can be called as server functions.

### Examples

```
[345] sm1_deRham([x^3-y^2*z^2,[x,y,z]]);
[1,1,0,0]
/* dim H^i = 1 (i=0,1), =0 (i=2,3) */
```

```
[287] phc(katsura(7)); B=map(first,Phc)$
[291] gnuplot_plotDots(B,0)$
```

### Authors

Castro-Jiménez, Dolzmann, Hubert, Murao, Noro, Oaku, Okutani, Shimoyama, Sturm, Takayama, Tamura, Verschelde, Yokoyama.

## 4 OX サーバを組み合わせて利用した例

具体的な数学的問題をもとに数学ソフトウェアの開発をするというのは一つの健全な開発手法であろう。

OX サーバの開発でも、そのような開発手法をとりたいと思っており、いろいろな数学的問題をさがしている。

その中の一つの問題は、多変数超幾何関数に関して何でも答えられるシステムの開発である。ここでは、多変数超幾何関数として、いわゆる GKZ hypergeometric system の解を考えている。GKZ hypergeometric system は  $n$  次元空間の点集合に付随してきまる連立線形偏微分方程式系である。 $n$  次元空間の点集合を考えるため、多面体の幾何を扱う必要があるし、連立線形偏微分方程式系を考えるため、 $D$  加群を扱う必要もある。GKZ system は affine toric ideal を含むため、affine toric ideal のシステムも必要である。関数の数値計算のためには、数値積分的な手法も重要であるが、これはまだ全く手をつけていない。GKZ hypergeometric system については、本 [4] を参照。

現在の OpenXM/src/asir-contrib のパッケージは、GKZ hypergeometric system に関して何でもこたえられるシステムを目標に、asir をクライアントとして、ox servers をいろいろくっつけたシステムである。

例を一つあげよう。dsolv\_starting\_term は正則ホロノミック系を cone 上で収束する多変数の級数で解くための関数の一つであり、級数展開の主部をもとめる。

1. 行列 (点配置)  $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 \end{pmatrix}$  に付随する GKZ hypergeometric system を関数 sm1\_gkz で求める。

```
[1076] F = sm1_gkz( [ [1,1,1,1,1],[1,1,0,-1,0],
```

```

[0,1,1,-1,0]], [1,0,0]]);
[[x5*dx5+x4*dx4+x3*dx3+x2*dx2+x1*dx1-1,-x4*dx4+x2*dx2+x1*dx1,
-x4*dx4+x3*dx3+x2*dx2,
-dx2*dx5+dx1*dx3,dx5^2-dx2*dx4],[x1,x2,x3,x4,x5]]

```

2. このシステム  $F$  の方向  $(1,1,1,1,0)$  における級数解の主部を求める.

```

[1077] A= dsolv_starting_term(F[0],F[1],[1,1,1,1,0])$
Computing the initial ideal.
Done.
Computing a primary ideal decomposition.
Primary ideal decomposition of the initial Frobenius ideal
to the direction [1,1,1,1,0] is
[[[x5+2*x4+x3-1,x5+3*x4-x2-1,x5+2*x4+x1-1,
3*x5^2+(8*x4-6)*x5-8*x4+3,
x5^2-2*x5-8*x4^2+1,x5^3-3*x5^2+3*x5-1],
[x5-1,x4,x3,x2,x1]]]

```

```

----- root is [ 0 0 0 0 1 ]

```

```

----- dual system is

```

```

[x5^2+(-3/4*x4-1/2*x3-1/4*x2-1/2*x1)*x5+1/8*x4^2
+(1/4*x3+1/4*x1)*x4+1/4*x2*x3-1/8*x2^2+1/4*x1*x2,
x4-2*x3+3*x2-2*x1,x5-x3+x2-x1,1]

```

3. 展開の主部 4 通りあり, それぞれを因数分解すると, 次のようになる. たとえば, 3 番目の  $[[1,1],[x5,1],[-\log(x1)+\log(x2)-\log(x3)+\log(x5),1]]$ , は,

$$x_5(-\log x_1 + \log x_2 - \log x_3 + \log x_5) = x + 5 \log \frac{x_2 x_5}{x_1 x_3}$$

から始まる級数解が存在することを意味する.

```

[1078] A[0];
[[ 0 0 0 0 1 ]]
[1079] map(fctr,A[1][0]);
[[[1/8,1],[x5,1],[log(x2)+log(x4)-2*log(x5),1],
[2*log(x1)-log(x2)+2*log(x3)+log(x4)-4*log(x5),1]],
[[1,1],[x5,1],[-2*log(x1)+3*log(x2)-2*log(x3)+log(x4),1]],
[[1,1],[x5,1],[-log(x1)+log(x2)-log(x3)+log(x5),1]],
[[1,1],[x5,1]]]

```

dsolv\_starting\_term では, kan/sml が D-加群の計算, asir が準素イデアル分解の計算, 解の計算を担当している. 解の展開の方向が何通り存在するかを調べるには, tigers を援用するとよい.

## 5 新しい数学ソフトウェアを OpenXM に追加するには?

研究上, 必要な計算の一部を実行するのに, OpenXM 非対応の数式処理システムが必要な場合がありうる. また, 普段使いなれているシステムでプログラムを書きたいということもある. そのような場合に, その数式処理システムを OpenXM 規約に対応させる方法を説明しよう. ある数式処理システムを新たに OpenXM サーバにするには, みつつのやり方が考えられる.

ひとつは, そのシステム自体に直接に変更を加えることである. これはもちろん, そのシステムのソースがなければ不可能である. この例としては, gnuplot, PHC, TiGERS がある.

ふたつめは, そのシステムを呼び出すような wrapper を用意することである. クライアントと wrapper との間は, OpenXM プロトコルで通信し, wrapper は OpenXM サーバとしてふるまう. 数式処理システムと wrapper との間はそのシステムの固有の通信プロトコルを用いる. この方法をとっているのは, Mathematica である. Mathematica の場合は, wrapper と Mathematica kernel との間は, MathLink を用いて通信している.

最後は, その数式処理システムが内蔵する言語で, OpenXM サーバを書くことである. Macaulay 2 は, この方法を用いている. ただし, 任意のファイルディスクリプタを読み書きする機能はなかったのが, M2 起動時に, 標準入出力はファイルディスクリプタ 3,4 に dup されている. (3,4 は OpenXM 規約で決められた特別なディスクリプタである)

## 6 OpenXM の入手方法

OpenXM の入手手順は公式ウェブページ <http://www.openxm.org/> に指示されているが, ここで少し補足しておく.

まず, OpenXM には安定版と開発版がある. 安定版にはリリース番号がつけられ, 原則的に匿名 ftp または Web で入手可能である. 公式の匿名 ftp サイトは <ftp://ftp.math.sci.kobe-u.ac.jp/pub/OpenXM/> である.

OpenXM の開発版はいま, 開発途上のソースそのものである. したがって, 稀ではあるが, 思わぬバグが入り込んでいたりコンパイルができない場合も有り得るので要注意である. ただし, 新しい機能は開発版にしか存在しないこともある.

開発版は CVS あるいは CVSup によって入手可能である.

## 6.1 匿名 CVS による入手

OpenXM プロジェクトではソフトウェアの開発, 保守に CVS を利用している. ユーザは, committer でなくとも, 匿名 CVS によって, ソースを入手できる.

まず, オペレーティングシステムに CVS がインストールされている必要がある. 現在の多くのシステムでは最初から CVS がインストールされているか, あるいは, バイナリパッケージで用意されている. もし, CVS がインストールされていなければ, <http://www.cvshome.org/> から入手してほしい.

さて, 以下の手順で, 匿名 CVS を利用できる. まず, 最初の一回は, パスワードを登録する手続きが必要である.

```
% setenv CVSROOT
      :pserver:anoncvs@kerberos.math.kobe-u.ac.jp:/home/cvs
% cvs login
```

ここで, パスワードを聞かれるので, anoncvs と入力する. これで, \$HOME/.cvspass にパスワードが記入されたはずである.

準備が終わったら, 実際に checkout を行おう.

```
% setenv CVSROOT
      :pserver:anoncvs@kerberos.math.kobe-u.ac.jp:/home/cvs
% cvs checkout OpenXM OpenXM_contrib OpenXM_contrib2
```

カレントディレクトリに OpenXM, OpenXM\_contrib, OpenXM\_contrib2 の3つのディレクトリができてはいるはずである. OpenXM のソースはこれらのディレクトリに含まれている.

## 6.2 CVSup による入手

CVSup はフリーソフトウェアであり,

<http://www.polstra.com/projects/freeware/CVSup/>

からソースまたはバイナリが入手できる.

CVSup では開発版の最新のソースのみを入手できる. CVS のように, 特定のバージョンを指定してダウンロードすることはできない.

さて, CVSup をインストールしたら, OpenXM のソースを展開するディレクトリと, CVSup の作業ディレクトリを決める. ここではそれぞれ, /home/ohara/openxm および /home/ohara/tmp としよう. これらのディレクトリはあらかじめ用意しておく. さて, 次の内容で openxm-supfile を作成する.

```

*default host=cvsup.math.sci.kobe-u.ac.jp
*default base=/home/ohara/tmp
*default prefix=/home/ohara/openxm
*default release=cvs tag=.
*default delete use-rel-suffix
*default compress
openxm
openxm-contrib
openxm-contrib2

```

あとはコマンドラインから次のように打ち込む.

```
% cvsup -g openxm-supfile
```

開発版の全てのソースが /home/ohara/openxm/ 以下にダウンロードされるはずである.

## 参 考 文 献

- [1] Cox, D., Little, J., O'Shea, D., *Ideals, Varieties, and Algorithms*, 2nd edition, Springer-Verlag, 1996.
- [2] Cox, D., Little, J., O'Shea, D., *Using Algebraic Geometry*, Springer-Verlag, 1998.
- [3] Huber, B., Sturmfels, B., A Polyhedral Method for Solving Sparse Polynomial Systems, *Mathematics of Computation*, **64** (1995), 1541–1555.
- [4] Saito, M., Sturmfels, B., Takayama, N., *Gröbner deformations of hypergeometric differential equations*. Algorithms and Computation in Mathematics, 6. Springer-Verlag, Berlin, 2000.
- [5] 野呂正行: 計算代数入門, Rokko Lectures in Mathematics, 9, 2000. ISBN 4-907719-09-4.  
<ftp://ftp.math.kobe-u.ac.jp/pub/OpenXM/Head/openxm-head.tar.gz>  
 のディレクトリ OpenXM/doc/compalg にこの本の  $\text{\TeX}$  ソースがある.
- [6] 高山信毅, 野呂正行: OpenXM の設計と実装 (OX-RFC-100), 2000/11/17.  
<http://www.math.sci.kobe-u.ac.jp/OpenXM/1.1.3/html/OpenXM-ja/OpenXM/>
- [7] 小原功任: エンジン起動プロトコル (OX-RFC-101 Draft).  
<http://www.math.sci.kobe-u.ac.jp/OpenXM/1.1.3/html/OX-RFC-101-ja/OX-RFC-101/>